CS 2100

AVL Trees
This week + next:

- HW due today - given website error, may submit tomorrow.
- Next HW: remove in a BST, remove in an AVL
- Tuesday & Wednesday next week
- tomorrow will be a work day (no lab)
- Today: office hours at 2pm (noon-2:30)
- Likely not be a lab next Tuesday - lecture instead
- Midterm 2: April 16
Last time:

Binary Search trees.

What's left - runtimes?

**Find**: $O(n) \rightarrow O(2^h)$

**Insert**:

**Remove**: $O(\text{height}(T))$

Alternative: $O(\text{height}(T))$
Today: How to make better?

Well, why were heaps $O(\log n)$ instead of $O(n)$?

**Goal**: How can we balance?

Ex: $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$
Balanced BSTs
Many kinds:
- Red-black trees: $O(1.4 \log_2 n)$
- Splay trees:
- AVL trees: $O(2 \log_2 n)$

Goal of them all: $O(\log_2 n)$
AVL trees:
Height balance property:
For every node $v$ in $T$, the heights of $v$'s children differ by at most 1.

$\Rightarrow$ max height $\leq 2 \lceil \log_2 n \rceil$
Ex: Insert (2) ← OK

Ex: Insert (47) ← NO!

Now: How can we mess this up?

Fix it! How? rotate (or pivot) 48 up
Consider the lowest node which fails the Height-Balance property. Call this \( z \).

Let \( Y \) be child with bigger height.
Let \( X \) be grandchild with larger height.

\[ \text{Fix!} \]

Move \( Y \) up at balance.
Another: insert (60) below.

Consider the lowest node which fails the Height-Balance property.

Let \( y \) be child w/ bigger height.

Let \( x \) be grandchild w/ larger height.

Promote \( x \).

Fix!
So: algorithm to insert:
- First run BST insert
- Then find lowest unbalanced node at deeper child/ grandchild.

Promote middle value
Fit: make x the root

Case #3
Key operation: \textit{pivot}

\begin{align*}
\text{pivot } (b) & \quad \text{pivot } (b) \\
\text{b} & \quad \text{a} \\
T1 & \quad a \\
T2 & \quad b \\
T3 & \\
\end{align*}
Then: implement w/ pivot!
Bigger example:

Insert: 1, 2, 3, 4, 5, 6, 7